

Labor Automatisierungstechnik - User Manual

Thimo Langbehn

10. Dezember 2013

Inhaltsverzeichnis

1	Kerberos	2
1.1	Voraussetzungen	2
1.2	Programme	2
1.3	Konfiguration	2
1.3.1	Linux	3
1.4	Verwendung	3
1.4.1	Linux	3
1.5	Fehlerbehebung	3
2	SVN	4
2.1	Voraussetzungen	4
2.2	Programme	4
2.3	Konfiguration	5
2.3.1	Subversion	5
2.3.2	Eclipse	5
2.4	Verwendung	5
2.5	Test	5
2.5.1	Linux	5
2.6	Migration von Alten Checkouts (Working Copy)	6
2.6.1	Linux	6
3	Andrew File System (AFS)	6
3.1	Voraussetzungen	6
3.2	Programme	6
3.3	Konfiguration	7
3.3.1	OpenAFS auf Debian	7
3.3.2	OpenAFS auf Gentoo	7
3.3.3	OpenAFS auf Ubuntu 8.04	7
3.3.4	OpenAFS auf Ubuntu 8.10	7
3.3.5	Manuell Linux Konfiguration	8
3.4	Test	8
3.4.1	Linux	8
3.5	Fehlerbehebung	9
3.5.1	Linux	9

4	Player/Stage	9
4.1	Voraussetzungen	9
4.2	Programme	10
4.3	Installation	10
4.3.1	Download	10
4.3.2	Update	10

1 Kerberos

Um in den vollen Genuss der Möglichkeiten eines Kerberos-Systems (Single Sign On) zu kommen muss man eine Kerberos-Implementierung installieren. Die Implementierung besteht im wesentlichen aus einer Kerberos Bibliothek und einigen Programmen zur Ticket(cache) Verwaltung.

1.1 Voraussetzungen

- Die Namensauflösung für die Domäne „**beuth-hochschule.de**“ muss funktionieren. Die korrekten Nameserver für die BHT sind:
141.64.3.55 (dnsmaster.beuth-hochschule.de)
141.64.3.40 (dnsslave.beuth-hochschule.de)
Der DNS-Server mit der IP 141.64.224.200 ist veraltet und soll nicht mehr verwendet werden. Er wird bei Kerberos zu Problemen führen.
- Der Host „**lat30.beuth-hochschule.de**“ muss erreichbar sein (**UDP Port 88**).

1.2 Programme

Linux Je nach Linux-Distribution haben die MIT Kerberospakete verschiedene Namen:

Debian krb5-user

Ubuntu krb5-user, krb5-config

Gentoo mit-krb5

1.3 Konfiguration

Die Bibliothek muss über eine Konfigurationsdatei eine Liste von sogenannten Realm Spezifikationen erhalten, um einer DNS Adresse einen Realm, und diesem einen KDC-Server zuzuordnen. Sie muss wie folgt abgeändert werden:

```
[libdefaults]
    default_realm = LAT.BEUTH-HOCHSCHULE.DE
    dns_lookup_kdc = true
[realms]
    LAT.BEUTH-HOCHSCHULE.DE = {
        admin_server = lat30.beuth-hochschule.de
        kdc = lat30.beuth-hochschule.de
    }
[domain_realm]
```

```
beuth-hochschule.de = LAT.BEUTH-HOCHSCHULE.DE
.beuth-hochschule.de = LAT.BEUTH-HOCHSCHULE.DE
[logging]
```

Die restlichen Realm-Beschreibungen (in der Sektion **[Realms]**) können bestehen bleiben.

Diese Datei ist insofern nicht sicherheitskritisch, als dass sie problemlos gelesen werden darf. Schreibzugriff sollte allerdings nur einem Administrator gestattet sein. Ansonsten könnte durch Manipulation der **kdc** und **admin_server** Adresse eine Spoofing-Attake eingeleitet werden.

1.3.1 Linux

Unter Linux liegt die **krb5.conf** ganz normal unter **/etc/**. Für Ihre Änderung werden root-Rechte benötigt.

1.4 Verwendung

Voraussetzungen für die Verwendung von Kerberos sind die oben beschriebenen Einstellungen, eine bestehende Netzwerkverbindung (Internet) und Verfügbarkeit des korrekten DNS-Servers. Weiterhin muss die Uhrzeit des Rechners korrekt eingestellt sein.

1.4.1 Linux

Zeitsynchronisation Unter Linux wird die Zeitsynchronisation am besten per ntp durchgeführt (**ntpdate pool.ntp.org; hwclock --systohc**), hierfür sind root-Rechte nötig.

Einloggen Ein Ticket kann mittels **kinit USER** gezogen werden. Sollte der lokale Login-Name mit dem gewünschten Kerberos Usernamen übereinstimmen, so kann der Name auch weggelassen werden. Nützliche Optionen für **kinit** sind **-f** (forwardable) und **-l 1d** (ein Tag Gültigkeit). Wenn die Anforderung erfolgreich war, kann das nun vorhandene Ticket mit **klist** angezeigt werden.

Passwort ändern Mit dem Programm **\$> kpasswd** kann das Kerberos Passwort geändert werden. Es wird genauso benutzt wie das **passwd** Programm.

1.5 Fehlerbehebung

1. Synchronisiere die Zeit mit einem Internet-Zeitserver. Sollte dies mit einer Fehlermeldung abbrechen, oder die Zeit nicht korrekt synchronisiert werden, so kann eine Ursache dafür in einer dazwischenliegenden Firewall liegen. In diesem Fall kann die Zeit auch manuell gestellt werden. Sie muss nur innerhalb einer Toleranz von 5 Minuten korrekt sein.
2. führe ein **\$> ping lat30.beuth-hochschule.de** aus, dabei sollte der Name in die korrekte IP (**141.64.75.95**) aufgelöst werden.

Funktioniert dieser Test nicht, dann arbeitet die DNS-Auflösung nicht richtig. Das kann an einem fehlerhaften Nameserver liegen, oder auch an überschreibenden Einstellungen des Clients bzw. alten Werten in dessen Puffer.

Eine andere IP (nicht 141.64.75.95) wird angezeigt Irgendein Nameserver oder Cache liefert die falschen Werte zurück. Überprüfe die hosts Datei. Diese findet sich unter Windows in **C:\Windows\System32\drivers\etc\hosts**, und unter Linux in **/etc/hosts**. Suche nach Einträgen der Domain „**beuth-hochschule.de**“. Diese sollten entfernt werden, sie sind nur nötig wenn der Nameserver keine korrekten Ergebnisse liefert.

Es wird gar keine IP angezeigt Die Domäne wird von dem lokalen Nameserver nicht richtig aufgelöst. Trage die folgenden Einträge in die oben genannte hosts Datei ein:

```
141.64.75.95  lat30.beuth-hochschule.de
```

3. Überprüfe deine Firewall, ob sie ausgehende Verbindungen zum **UDP Port 88** von **141.64.75.95** zulässt. Ein Test in einer weniger restriktiven Umgebung kann Unklarheiten bezüglich anderer, dazwischen liegender Umgebungen beseitigen).

2 SVN

Auf dem Server können eine beliebige Anzahl von SVN Repositories betrieben werden, sei es für den ausschließlich eigenen Gebrauch, oder für eine Gruppe.

Zugriff auf die SVN Repositories erfolgt direkt über das svn Protokoll:

2.1 Voraussetzungen

- Kerberos muss eingerichtet sein.
- **lat30.beuth-hochschule.de** muss erreichbar sein.

2.2 Programme

Windows 7 64Bit oder Vista 64Bit Hierfür existiert zwar bereits eine TortoiseSVN 64Bit Version:

<http://downloads.sf.net/tortoisesvn/>

Allerdings fehlt derzeit (März 2010) noch das benötigte saslGSSAPI.dll in einer 64Bit Version. Somit kann SVN als 64Bit Lösung derzeit nicht mit Kerberos unter Windows verwendet werden. Für die 32-Bit Lösung muss dann auch Kerberos in 32-Bit installiert sein.

Linux Allgemein Um unter Linux mittels GSSAPI (Kerberos) und SVN die Repositories anzusteuern wird ein subversion client der Version 1.5 oder höher mit SASL Support benötigt. Das SASL GSSAPI Plugin muss ebenfalls installiert sein.

Debian Etch Das **subversion** Paket ist zu alt, bis jetzt ist keine Alternative bekannt.

Debian Lenny Das **subversion** Paket ist ok, es wird zusätzlich noch das **libsasl2-modules-gssapi-mit** Paket benötigt.

Gentoo dev-util/subversion mit einer Version über 1.5.0 mit dem **sasl** USE-Flag funktioniert. Weiterhin muss **dev-libs/cyrus-sasl** mit dem USE-Flag **kerberos** kompiliert sein.

Ubuntu >=8.10 (Intrepid Ibex) Das **subversion** Paket ist ok, es wird zusätzlich noch das **libsasl2-modules-gssapi-mit** Paket benötigt.

2.3 Konfiguration

2.3.1 Subversion

Für den einfachen (Client) Zugang bedarf es unter Linux keiner weiteren Konfiguration.

2.3.2 Eclipse

Um SVN aus Eclipse heraus nutzen zu können, beispielsweise mit Subclipse (<http://subclipse.tigris.org/>), muss die Kerberos-Authentifizierung für diese Plugins nachgerüstet bzw. aktiviert werden. Für Linux und Windows hält das MIT eine Anleitung vor: <https://wikis.mit.edu/confluence/display/ISTMAP/Integrating+MyEclipse+and+Subversion>. Während der erste Absatz MyEclipse-spezifisch ist, kann der Rest auch bei anderen Eclipse-Integrationen Anwendung finden.

2.4 Verwendung

Vor dem Zugriff auf Kerberos-basierte SVN Repositories muss ein Ticket angefordert werden. Je Nach Anmeldeclient wird, sollte dieses zum Zugriffszeitpunkt nicht vorhanden sein, die Eingabe der User-Credentials gefordert. Dies ist jedoch nur einmalig im Gültigkeitszeitraum des Tickets notwendig.

Wenn ein Ticket vorhanden ist, können vorhandenen Repositories unter der URL **svn://lat30.beuth-hochschule.de/REPOSITORY** angesprochen werden. **REPOSITORY** muss hierbei natürlich durch einen gültigen namen ersetzt werden.

2.5 Test

Als Beispiel und zum Experimentieren kann das **sandbox** Repository verwendet werden, die dazugehörige URL lautet **svn://lat30.beuth-hochschule.de/sandbox**.

Dazu bedarf es folgender Schritte:

2.5.1 Linux

Öffne eine Konsole und gebe **svn co svn://lat30.beuth-hochschule.de/sandbox** ein.

Der checkout sollte erfolgreich durchgeführt werden können, in dem sandbox Verzeichnis sollten nun einige Dateien und Unterverzeichnisse zu finden sein.

2.6 Migration von Alten Checkouts (Working Copy)

2.6.1 Linux

Hier wechselt man in das Wurzelverzeichnis des checkout-Verzeichnisses.

Dann kann mit **svn info | grep URL** die ursprüngliche, alte URL des Repositories ermittelt werden. Zum Beispiel *svn+ssh://lat30.beuth-hochschule.de/sandbox*.

Die neue URL kann nun durch Ersetzen des ersten Teils (bis zum Repository-Namen mit **svn://lat30.beuth-hochschule.de/** gebildet werden. Dies ergibt in diesem Beispiel also *svn://lat30.beuth-hochschule.de/sandbox*.

Immer noch im Wurzelverzeichnis des Checkouts kann mit dem Kommando **svn switch -relocate URL-ALT URL-NEU** der Checkout an die neue Position auf dem Server angepasst werden. Das gibt in dem Beispielfalle also:

```
svn switch -relocate \  
svn+ssh://lat30.beuth-hochschule.de/sandbox \  
svn://lat30.beuth-hochschule.de/sandbox
```

3 Andrew File System (AFS)

3.1 Voraussetzungen

- Kerberos muss eingerichtet sein.
- Ein entsprechender Account muss eingerichtet sein.
- **lat30.beuth-hochschule.de** muss erreichbar sein.

3.2 Programme

Debian Etch und Lenny Die Pakete **openafs-client**, **openafs-krb5** sowie **openafs-modules-source**. Die installation des Moduls sollte über den module assistant (**module-assistant**) vorgenommen werden.

Gentoo Der stable **openafs-kernel-1.4.6** und **openafs-1.4.6** kompiliert (zumindest auf 64-Bit Systemen) nicht gegen einige Kernel Sourcen (2.6.25). Statt die unstable Version zu nehmen sollte hier ein anderer Kernel verwendet werden (2.6.24). Die unstable Version dieses ebuild hat sich auf Intel und AMD 64-Bit systemen als wirklich instabil erwiesen und führte bisher immer nach kurzer Nutzung zu einem Einfrieren der entsprechenden Prozesse.

Ubuntu 8.04 LTS (Hardy Heron) Das mitgelieferte openafs-kernel Paket lässt sich nicht kompilieren (siehe Bug #129480).

Ubuntu 8.10 (Intrepid Ibex) Benötigt die Installation der Pakete **openafs-client**, **openafs-krb5** sowie **openafs-modules-source**. Zusätzlich ist es ratsam, den module assistant (**module-assistant**) zum installieren des Modules zu benutzen.

Ubuntu 10.04 (Lucid Lynx) **openafs-client** und **openafs-krb**. Dies führt zu einer automatischen Installation des openafs-Kernelmoduls (**openafs-modules-dkms**) über den module assistant. Demzufolge werden, wenn

noch nicht vorhanden, auch eine ganze Reihe von Werkzeugen mitinstalliert. Das Module wird nach Kernel-Updates automatisch neu erstellt.

3.3 Konfiguration

3.3.1 OpenAFS auf Debian

Als erstes muss das Modul für Debian kompiliert werden. Falls er noch nicht installiert ist, dazu mit `$> apt-get install module-assistant` einspielen.

Dann mit `$> m-a prepare openafs` das herunterladen von allen zum kompilieren des opeafs-Kernelmoduls notwendigen Paketen veranlassen. Hiernach kann mit `$> m-a a-i openafs` das entsprechende module erstellt werden.

Anschließend noch mit `$> apt-get install openafs-client openafs-krb5` den OpenAFS client für Linux installieren und konfigurieren.

Der Zellename für LAT.BEUTH-HOCHSCHULE.DE ist **lat.beuth-hochschule.de**, der primäre File- und Volumeserver ist **lat30.beuth-hochschule.de**. Der Client kann ruhig mit dem Rest des Systems gestartet werden.

3.3.2 OpenAFS auf Gentoo

Ein `$> emerge net-fs/openafs net-fs/openafs-kernel` mit dem aktivierten USEFLAG **kerberos** sollten für die Installation ausreichen. Dabei sollten beide pakete nicht in `/etc/portage/package.keywords` aufgeführt sein.

Wenn es hierbei zu einem Compile-Fehler kommt wird vermutlich eine inkompatible Version des Linux-Kernel verwendet. In diesem Falle kann für die Version **1.4.6_p20080222** der letzte Kernel der Version **2.6.24** verwendet werden. In Jedem Falle NICHT die unstable (maskierte) Version der openafs Pakete wählen.

Anschließend kann der OpenAFS client noch in die Autostartliste eingetragen werden mit `$> rc-update add openafs-client default`.

3.3.3 OpenAFS auf Ubuntu 8.04

In Ubuntu 8.04 (Hardy Heron) kompiliert das OpenAFS Paket der Version 1.4.6.dfsg1-2 nicht gegen den verwendeten Kernel 2.6.24-21-rt, wie in Bug #129480 beschrieben (<https://bugs.launchpad.net/ubuntu/+source/openafs/+bug/129480>). Hintergrundinformationen zu dem Begriff Tainted-Kernel und den GPLONLY Flags findet man unter <http://www.kernel.org/pub/linux/docs/1kml/#s1-18>.

Die Paketversion 1.4.7.dfsg1- behebt diesen Fehler, ist aber erst in Ubuntu 8.10 (Intrepid Ibex) enthalten (Release am 30.10.2008).

3.3.4 OpenAFS auf Ubuntu 8.10

Als erstes muss das Modul für Ubuntu kompiliert werden. Falls es noch nicht installiert ist, dazu mit `$> aptitude install module-assistant` einspielen.

Dann mit `$> m-a prepare openafs` das herunterladen von allen zum kompilieren des opeafs-Kernelmoduls notwendigen Paketen veranlassen. Hiernach kann mit `$> m-a a-i openafs` das entsprechende Modul erstellt werden.

Anschließend noch mit `$> aptitude install openafs-client openafs-krb5` den OpenAFS client für Linux installieren und konfigurieren.

Der Zellename ist **lat.beuth-hochschule.de**, der primäre File- und Volumeserver ist **lat30.beuth-hochschule.de**.

Der Client kann ruhig mit dem Rest des Systems gestartet werden, jedoch ist hier noch eine kleine Korrektur vonnöten, wie auch aus folgendem Bugreport zu entnehmen ist: <https://bugs.launchpad.net/ubuntu/+source/openafs/+bug/318605>. Das Setzen der Option **AFS_DYNROOT=true** in der Datei **/etc/openafs/afs.conf.client** ermöglicht das fehlerfreie Funktionieren von AFS.

3.3.5 Manuell Linux Konfiguration

Sollten diese Einstellungen aufgrund des installtionsmechanismus nicht abgefragt werden (Gentoo, Debian mit entsprechenden debconf-Einstellungen), so muss der Zellenname (**lat.beuth-hochschule.de**) in **/etc/openafs/ThisCell** eingetragen werden, z.B. mit **\$> echo 'lat.beuth-hochschule.de' > /etc/openafs/ThisCell**. Weiterhin muss der Fileserver in die **/etc/openafs/CellServDB** eingefügt sein.

Der dazugehörige Ausschnitt:
>lat.beuth-hochschule.de
141.64.75.95# lat30.beuth-hochschule.de

Eine passende Datei kann auch direkt von <http://lat.beuth-hochschule.de/downloads/CellServDB> heruntergeladen werden.

Anschließend noch den Client starten (mit **\$> /etc/init.d/openafs-client start**) und es kann losgehen. Der AFS-Namensraum wird normalerweise in **/afs** gemountet.

3.4 Test

3.4.1 Linux

1. Mit **\$> kinit** ein Kerberos Ticket für den User beschaffen.
2. **\$> aklog** aufrufen um mit Hilfe des Kerberos Tickets ein AFS Token zu erwerben.
3. Versuchen in das Persönliche Verzeichnis im AFS zu wechseln. Dieses befindet sich unter **/afs/lat.beuth-hochschule.de/user/USERNAME**, USERNAME ist hier durch den eigenen User zu ersetzen. Demnach also **\$> ls /afs/lat.beuth-hochschule.de/user/USERNAME**. Der Zugriff auf das AFS kann beim ersten Mal einige Momente brauchen (auch schon bei der Tab-Expansion des Pfades).

Start Allgemein gilt, erst ein Kerberos Ticket anfordern (**\$> kinit**), dann ein AFS-Token (**\$> aklog**).

Status Die AFS-Tokens können mit **\$> tokens** angezeigt werden. Die Kerberos Tickets mit **\$> klist**.

Stop Die AFS-Tokens werden durch **\$> unlog** gelöscht, die Kerbero-Tickets mittels **\$> kdestroy**.

3.5 Fehlerbehebung

- Die Systemzeit des eigenen Rechners muss korrekt sein. Eine Differenz von mehr als drei Minuten wird das Anfordern von Kerberos-Tickets oder AFS-Tokens verhindern.
- Das KDC und der AFS Fileserver (lat30.beuth-hochschule.de) müssen erreichbar sein.

3.5.1 Linux

Allgemeines Vorgehen:

1. Der OpenAFS Client muss gestartet sein (Gentoo fügt ihn z.B. nicht zum Autostart hinzu).
2. Ein aktuelles und gültiges Kerberos-Ticket muss vorhanden sein. Einmal löschen (`$> kdestroy`) und neu anfordern (`$> kinit`). Mit `$> klist` überprüfen. Dort sollte nun nur ein Eintrag für den Service principal **krbtgt/LAT.BEUTH-HOCHSCHULE.DE@LAT.BEUTH-HOCHSCHULE.DE** aufgeführt sein.
3. Ein neues AFS-Token anfordern (`$> aklog`). Anschließend sollte der die Ausgabe von `$> tokens` genau einen Eintrag für **tokens for afs@LAT.BEUTH-HOCHSCHULE.DE** enthalten, mit einem Ablaufdatum (Expires ...) in der Zukunft.
4. AFS verwenden.

Spezielle Fehlerfälle:

aklog bleibt ununterbrechbar stehen Das ist typischerweise der Fall wenn das Openafs-Modul Fehler aufweist und die Prozesse nicht bedient. In diesem Falle hilft kurzfristig nur der Neustart des Rechners und langfristig das austauschen des Moduls gegen eine stabile Version.

aklog: Couldn't determine realm of user:)**aklog: unknown RPC error (...) while getting realm**
Es ist kein Kerberos Ticket vorhanden. Bitte mit `$> kinit` nachholen.

cannot open directory /afs/lat.beuth-hochschule.de/user: Permission denied
Offensichtlich fehlt das AFS-Token, mit `$> tokens` prüfen und mit `$> aklog` anfordern.

aklog: Couldn't get lat.beuth-hochschule.deAFS tickets: aklog: unknown RPC error (...) while
Hier stimmt vermutlich die Systemzeit nicht mit der Zeit auf dem Fileserver überein. Mit `$> ntpdate pool.ntp.org` kann die Zeit angeglichen werden.

4 Player/Stage

4.1 Voraussetzungen

- AFS muss eingerichtet sein
- Subversion muss installiert sein

4.2 Programme

Ubuntu 10.04 Benötigt das **git-core** Paket.

4.3 Installation

4.3.1 Download

Zuerst wird das Master-Repository kopiert, dann der passende Zweig gewählt, dessen Abhängigkeiten initialisiert und heruntergeladen. Anschließend werden player und stage gebaut und installiert.

Listing 1: Setup

```
git clone /afs/lat.beuth-hochschule.de/service/git/robots.git
cd robots
git checkout --track -b stage origin/stage
git submodule init
git submodule update
./scripts/prepare_ubuntu-10.04.sh
cd src
make player
make stage
```

Auf einem 32 Bit System muss noch die `/etc/ld.so.conf.d/playerstage.conf` editiert werden (erste Zeile entfernen).

4.3.2 Update

Um den Quellcode zu aktualisieren führen Sie (im robots Verzeichnis) **git pull** und anschließend **git submodule update** aus.